FoRCES protocol as a solution for interaction of control and forwarding planes in distributed routers

Ivo Kovačević[1], *dipl.ing.*

Abstract **- This paper provides an analysis of the communication protocol used between the control and forwarding network elements in the distributed router specified by the IETF FoRCES working group. Based on this analysis and NPF MPLS API, it proposes a model for MPLS LSP creation procedure applicable to the distributed router in FoRCES environment.**

Key words **- control plane, distributed router, FoRCES, forwarding plane, MPLS, protocol.**

I. INTRODUCTION

As Internet traffic grows in volumes and new services, like VoIP, Deep Packet Inspection or Intrusion Detection&Prevention emerge IP devices (e.g. routers, firewalls) should meet a number of demands in order to be able to effectively cover for the coming market requirements. One of the demands is to provide for a scalable architecture where processing capacity can be cost-effectively increased. Another demand is to provide for open and programmable platforms where new services can easily be introduced. One approach to addressing these requirements is the introduction of distributed router platforms, where in general routing and forwarding functionalities are physically separated, realized on dedicated platforms and communicating over a network through well defined interfaces. Distributed routers appear as a single system to the outside world. They introduce higher availability, scalability and flexibility of the router system, but on the other side cost in terms of delay can be incurred because of internal communication overhead. Concerning the programmability and openness of interfaces that should allow for dynamic creation of new services several approaches have been taken so far like IEEE PIN1520 standard [1], NPF (Network Processing Forum) APIs [2] and *Netconf* protocol [3]. These approaches have standardized software APIs in an open and layered fashion. *Netconf* defines a standardized XML based protocol for router configuration. On the other hand, Forwarding and Control Element (FoRCES) IETF working group [4] has defined a distributed router architecture, based on the physical separation of routing and forwarding tasks and specified a protocol for the communication among internal elements.

One analysis of distributed routers' implementations based on software routers and Netlink protocol and aligned with the FoRCES framework [5-6] has been given in [7]. These early implementations used the FoRCES architecture but didn't implement the FoRCES protocol as defined by the FoRCES working group.

The goal of this paper is based on the analysis of the specifics of the FoRCES protocol, to propose a solution for MPLS LSP (Label Switched Path) creation procedure in the distributed router.

The paper is organized as follows: Section II describes the architecture of the distributed router aligned with the IETF FoRCEs specification; Section III gives an overview of the FoRCES protocol; Section IV focuses on the analysis of transport layer requirements; Section V presents the proposal for MPLS LSP creation procedure and relevant control-data plane information exchange; Section VI contains the conclusion.

II. DISTRIBUTED ROUTER ARCHITECTURE

From the functionalities perspective IP router can be divided in two logical *planes:* the forwarding plane and the control plane. Forwarding plane deals with the forwarding and processing of each packet that enters and exits the router via its physical interfaces. The control plane runs control protocols (like routing protocols: OSPF, IS-IS, BGP, signaling protocols: RSVP, LDP, etc.) and provides forwarding, control and management decisions to the forwarding plane. In the distributed approach these two planes are generally implemented on different hardware platforms, physically separated and interconnected via L2/L3 network to enable internal communication.

ForCES IETF Working Group is aiming to standardize open, programmable distributed network architecture. The standardization also includes the specification of the protocol for communication between control and forwarding plane in the router. FoRCES working group has produced RFC 3654 [5] and RFC 3746 [6] that define the requirements and the architecture framework. Protocol for control-data plane interaction on top of transport layer is specified in [8]. Fig. 1, shows a distributed router with the elements' names aligned with the FoRCES terminology as well as standardized interfaces among them.

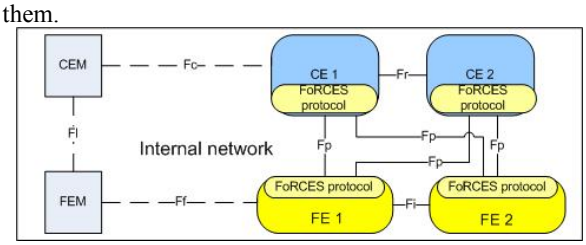

Fig. 1. *ForcES architecture.*

[1] Ivo Kovačević, Ericsson d.o.o, Belgrade, Serbia (email: ivo.kovacevic@ericsson.com)

The router *control plane* is instantiated in Control Elements (CEs) and the *forwarding plane* in Forwarding Elements (FEs). CE functions are typically performed in software running on general-purpose processors, while FEs can be based on different hardware platforms such as ASICs, FPGAs, network processors and general-purpose processors.

The internal network interconnects CEs and FEs. The purpose of the internal network is to carry control and data traffic between the elements. It can be built by utilizing different L1/L2 technologies (e.g. L2 Gigabit Ethernet switches). In this paper it is treated as a general L3 IP network, which is aligned with available implementation proposals [9-11].

In the distributed router IP data packets are forwarded on the data path among FEs interfaces using the Fi interface, but CE control traffic, routing protocol packets or network management packets are forwarded from the FE to the CE encapsulated in the FoRCES messages over the Fp interface.

FE internal architecture is standardized in ForCES FE Model [12]. The Resources in an FE are instantiated in Logical Function Blocks (LFBs), each of which has specific function in packet forwarding chain. Examples of LFBs are classifier LFB, scheduler LFB, IPv4 forwarder LFB, MPLS forwarder LFB. Multiple LFBs are interconnected via datapaths to form an LFB topology in FE, so that the FE can carry out a complex process on packets forwarding purpose. CE is responsible for the management of LFBs in FE. The management of LFBs includes the configuration and inquiry of LFB attributes, capabilities, or events. ForCES protocol makes it possible for CE to dynamically manage the LFBs such as to add/remove/modify some LFBs, the attributes, and the associated LFB topology. Manageability of FEs by CEs provided in this model is at much higher level than the manageability in present commercial routers aiming to provide the possibilities to configure new services in an open and simple way.

III. FORCES PROTOCOL OVERVIEW

FoRCES protocol is defined in two layers, Presentation layer (PL) and Transport Mapping Layer (TML) [8]. The PL is responsible for maintaining the association of FE or CE to an NE. An FE uses the PL to transmit various subscribed-to events to the CE as well as to respond to various status requests issued from the CE PL. The CE configures both the FE and associated LFBs' operational parameters using the PL.

The TML transports the PL messages. The TML is where the issues of how to achieve transport level reliability, congestion control, multicast, ordering, etc. are handled. Section IV discusses TML implementation options in more detail. FoRCES protocol involves two phases, pre-association phase and post-association phase.

Pre association phase is used for the *Fp* interface setup and initial discovery of CEs and FEs. Usually, it will be implemented by reading a static configuration from a file. At the completion of this stage both sides should know which NEs they belong i.e. which CE/FE will be associated with and have all the necessary protocol parameters(e.g. timers).

In the post association phase, FE and CE actually communicate with each other using the FoRCES protocol. This phase contains two stages: Association Setup Stage and Established stage. In the *Association Setup* stage FE attempts to join a previously configured CE. If it is granted, capability exchange can happen and CE can send FE initial configuration. In the *Established* stage the FE is queried and updated by CE. FE also sends asynchronous event notifications to the CE and heartbeats. This phase is kept until *Association* is torn down or connectivity is lost. All ForCES protocol messages operate on LFB instances. It means that the configurability of a router should be expressed in terms of LFB architecture. In the same way, by using special dedicated LFBs (FE Protocol LFB and FE Object LFB), ForCES protocol itself can be configured.

On the FoRCES protocol message level, following types of messages are defined: Association messages (Setup, Setup Response, Teardown), Configuration messages (Config, Config Response), Query messages (Query, Query response), Event Notification, Packet Redirect and Heartbeat messages. An example message exchange from the association setup to the teardown phase among FE and CE is shown in Fig.2.

In terms of message format all messages have a common header where basically message type, Source ID and Destination ID of the communicating elements are defined followed by the message body. Message body consists of one or more top level TLV fields which contain sub-TLVs depending on the type of the message type and operations to be performed.

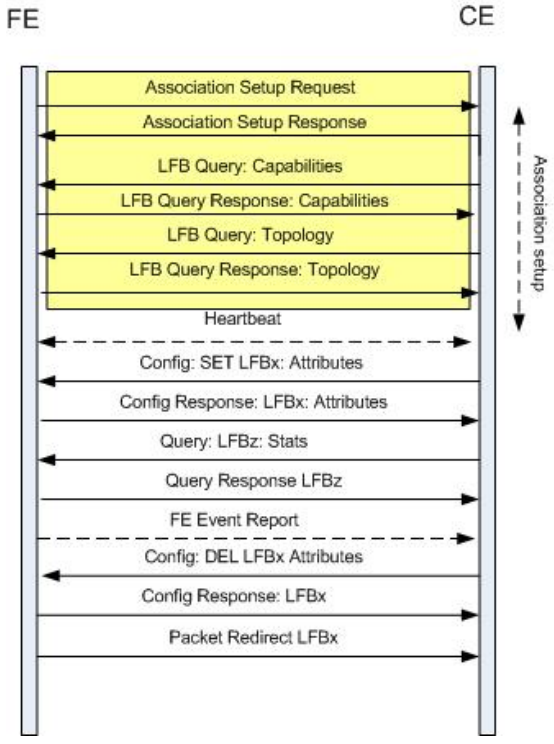

Fig. 2. FoRCES protocol : CE-FE message exchange.

IV. TRANSPORT LAYER

One aspect of the communication among network elements in the distributed router concerns the choice of the transport protocol. Without deeper analysis UDP can be a choice since it is a simple solution and it is also appropriate for IP multicast of messages towards a number of FEs. However, there are different requirements for the internal communication. For example, some information like route updates require reliable transport and some like heartbeat prefer low latency. Generally, TCP is appropriate to be used for the transport of control messages between CE and FE, where UDP as a more aggressive and efficient protocol (no session establishment overhead) is more appropriate for data transport (e.g. routing protocol messages, heartbeat messages). As proposed in [13] certain measures should be taken to prevent for UDP DoS. UDP does not have built-in congestion control mechanisms and therefore can consume all available bandwidth. Therefore an algorithm is proposed, so to define UDP-TCP pairs, where UDP stream will be controlled with reference to TCP stream concerning its congestion status. In other words, TCP traffic queue is prioritized over UDP and inherently ForCES protocol control traffic is protected from bandwidth consumption by data messages transported via UDP.

Another solution for the transport layer protocol proposed is SCTP (Stream Control Transmission Protocol) [14]. SCTP [15] can be understood as one common end-to-end protocol that is able to satisfy all aspects of a transport requirements specified for ForCES protocol. SCTP has some common features of both TCP and UDP. Like TCP, it provides ordered, reliable, connection-oriented, flow-controlled, congestion controlled data exchange. Unlike TCP, it does not provide byte streaming and instead provides message boundaries. Like UDP, it can provide unreliable, unordered data exchange but it does not provide multicast support. In addition it provides a number of advantageous services which TCP and UDP cannot provide. One of the most important is multi-homing, where SCTP provides the possibility to use multiple destination IP addresses to communicate with a peer. Built-in heartbeat is also very useful as it provides mechanisms to check the reachability of the peer. Multi-streaming feature provides for the transport of independent application streams over the same transport connection (socket). It also offers different "channels", realized as separate SCTP sockets offering the possibility to use reliable or non-reliable transport as well as built in congestion control mechanism. Mapping of ForCES messages to SCTP channels is presented in [14]. Security is improved in SCTP compared to TCP as it uses a 4-way handshake mechanism in contrast to TCP which uses 3-way handshake. This approach makes SCTP more secure against DoS SYN attacks.

Therefore SCTP will certainly be the protocol of choice for future mature FoRCES implementations, as it brings integrated mechanisms in one transport protocol that can fully cover for FoRCES protocol needs. In the meantime experimental and early implementations [9-11] are coming with TCP and/or UDP protocol implemented on the transport layer. This is considered as a straightforward approach, using protocols available in all common operating systems.

V. MPLS LSP CREATION: PROCEDURE PROPOSAL

Different proof of concepts of a FoRCES router addressed the implementation of the basic functionalities of a router, namely interface configuration, route distribution, IP forwarding [9-11], SNMP support [16]. Cross-analysis of some of the early implementations is given in [7].

One of the functionalities that should be supported in the modern router is Multiprotocol Label Switching (MPLS) [17] . Compared to IP routing MPLS brings less-complex routing process relying on label switching instead of IP prefix lookup. It also brings Traffic Engineering features what made MPLS become a prominent technology on top of which modern service provide networks are built up. Basic concept of MPLS is packet switching based on the MPLS labels. MPLS label is a fixed-length (20-bit header) added at the front of a packet. Labels are used as lookup indexes in forwarding table, so once a packet enters MPLS network, IP route lookup is no longer performed in each router in order to decide on which interface to route a packet. MPLS labels are distributed in MPLS network by a signaling protocol like LDP [18] or RSVP [19]. These signaling protocols are implemented in the control plane, whereas the switching of packets based on label information is handled in the forwarding plane (line cards).

Therefore, in the distributed router we need a sort of mechanism to distribute the label information from the control to the forwarding plane.

The proposal for MPLS LSP creation procedure described below refers to the software architecture of the distributed router shown in Fig.3. This architecture is aligned with the recent efforts [16]. The proposal is based on the combination of a FoRCES protocol and NPF MPLS API [20].

NPF has designed a service API for MPLS that provides a set of standardized functions, data structures and communication mechanisms (e.g. callback functions). These functions provide an interface to the application level software (e.g. LDP implementation). In the architecture shown in Fig.3, Functional MPLS API is also present and serves the function of adapting MPLS function calls and data into FoRCES data model structures, in this case MPLS LFB data formats. This layer should call FoRCES API in order to generate appropriate messages in the FoRCES communication.

Message flow for a process of one LSP creation is also shown in Fig 3. After registration of a callback function, application layer (e.g. LDP application instance) should make API function call in order to create an LSP by providing a number of LSPs to be created and an array of parameters for each of them. Once these parameters are mapped into FoRCES data structures (relevant LFB), *Config* (SET) message will be sent by the FoRCES PL layer over internal network towards the FE. This message is received and parsed by the FoRCES protocol layer in FE, and the create LSP command is conveyed to the FE low-level software. After the LSP has been created in FE, FoRCES PL layer responds with *Config Response* (SET RESPONSE) message. When this message is received in

CE, completion callback function is called to inform the application layer about the result of the command issued and provide return parameters if any. Proposed function calls and FoRCES messages are shown in Table 1.

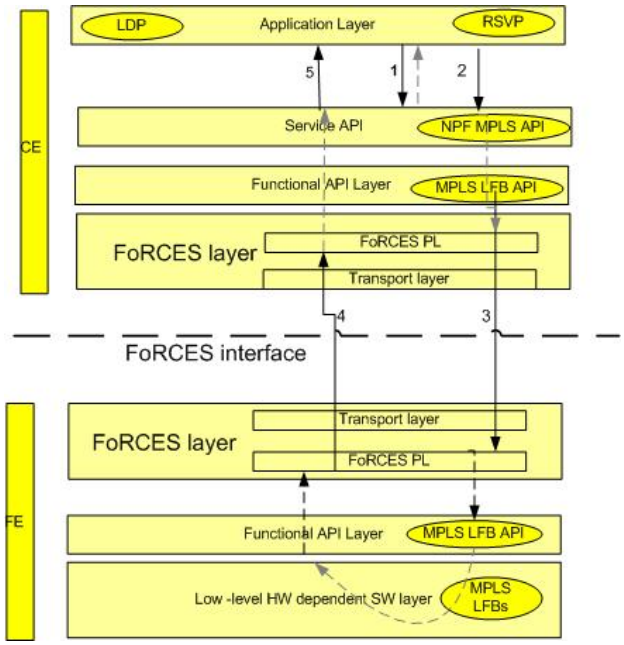

Fig. 3. Message flow: LSP creation in a distributed router.

TABLE 1: FUNCTION CALLS AND FORCES MESSAGES FOR LSP SETUP.

#	*Message*	*Description*
1	NPF_MPLS_Register (callbackFunc, *callbackHandle)	RSVP/LDP registers its callback function to receive response
2	NPF_MPLS_LSP_EntryCreate (callbackHandle,.., nMplsLsp, **mplsLspArray);	Calling function to create one MPLS LSP entry.
3	CONFIG, SET, LFB= MPLSx, ID=LSPx, Data= <LSP data>	FoRCES Config message, with SET operation, identifying MPLS LSP LFB
4	CONFIG RESPONSE, SET RESPONSE, Result TLV = E-SUCCESS, Data= LSPx	FoRCES Config Response message, successful creation of a label
5	NPF_MPLS_CallbackFunct_t (callbackData=LSPx, Success)	RSVP receives an answer that LSP is successfully created in the forwarding plane

VI. CONCLUSION

This paper gives the analysis of the protocol specified by FoRCES working group for internal communication between control and forwarding plane elements in the distributed router. It also makes a comparison of the transport layer options. In the transport layer SCTP is seen as a protocol to be used for future mature solutions. TCP/UDP with some congestion control mechanisms implemented is also acceptable and has the advantage of being widely available in common operating systems. FoRCES protocol will be leveraged in the distributed router environments to provide support for different router applications like IP routing, SNMP, MPLS switching. Based on the FoRCES protocol analyzed and NPF MPLS API, the paper describes one approach to solving the problem of MPLS protocol configurability in the distributed router, treating a case of configuring one LSP. Further studies in the area that will be focused on the definition of the full scope of MPLS distributed router parameterization based on FoRCES protocol and NPF MPLS API .

VII. LITERATURE

[1] J. Biswas, et al., "The IEEE P1520 Standards Initiative for Programmable Net-work Interfaces", *IEEE Communications, Special Issue on Programmable Net-works*, Vol. 36, No 10, October, 1998.

[2] David M. Putzolu, Network Processing Forum Software Work Group, "Software API Framework Implementation Agreement", 2002.

[3] Rensink Enns, Ed., "NETCONF Configuration Protocol", *RFC 4741*, December 2006.

[4] ForCES IETF Working group, URL= \ http://www.ietf.org/html.charters/forces charter.html.

[5] H. Khosravi, T. Anderson , "Requirements for Separation of IP Control and Forwarding", *RFC 3654*, Nov.2003.

[6] L. Yang, R. Dantu, T. Anderson, R. Gopal , "Forwarding and Control Element Separation (ForCES) Framework", *RFC 3746*, April 2004.

[7] I.Kovacevic, "Analysis of Control and Forwarding Plane Communication within Distributed Routers", *ETRAN2009 Conference*, Vrnjacka Banja, Serbia, 2009.

[8] A. Doria, et.al, "ForCES Protocol Specification", *Internet-Draft*, March 2009.

[9] O. Hagsand, M. Hidell, P. Sjödin, "Design and Implementation of a Distributed Router", *IEEE International Symposium on Signal Processing and Information Technology*, 2005.

[10] R. Bolla, R. Bruschi, G. Lamanna and A. Ranieri, "Beyond Single-Box SW Router Architectures*", IEEE Workshop on High Performance Switching and Routing (HPSR)*, 2009.

[11] Wang WM, Dong LG, Zhuge B , "Analysis and implementation of an open programmable router based on forwarding and control element separation", *Journal of Computer Science and Technology* 23(5), Sept. 2008

[12] J. Halpern, J.Hadi Salim, "ForCES Forwarding Element Model", *Internet-Draft*, October 2008.

[13] W.M.Wang, L.G.Dong, et al., "TCP and UDP based ForCES Protocol TML over IP Networks" , *Internet-Draft*, March 2007.

[14] J.Hadi Salim, K.Ogawa, "SCTP based TML (Transport Mapping Layer) for ForCES protocol", *Internet-Draft*, September 2009.

[15] R.Stewart, "Stream Control Transmission Protocol", *Internet-Draft*, September 2007.

[16] R.Jin, W.Wang, "An Efficient Way to Support SNMP in the ForCES Framework", *Communication Systems, ICCS 2008, 11th IEEE Singapore International Conference*, 2008.

[17] E.Rosen, A. Viswanathan et.al, "Multiprotocol Label Switching Architecture" , *RFC 3031*, January 2001.

[18] Andersson, I. Minei, "LDP specification", *RFC 5036 (RFC 3036)*, October 2007.

[19] L.Braden, L.Zhang, "Resource Reservation Protocol (RSVP)", *RFC 2205*, 1997.

[20] M. Srinivasan, R. Haddad**, "**MPLS Forwarding Service APIs with Diffserv and TE Extensions Implementation Agreement, *Network Processing Forum*, 2003.